Institute for Advanced Simulation

Wavelets and Their Application for the Solution of Poisson's and Schrödinger's Equation

Stefan Goedecker

published in

Multiscale Simulation Methods in Molecular Sciences,
J. Grotendorst, N. Attig, S. Blügel, D. Marx (Eds.),
Institute for Advanced Simulation, Forschungszentrum Jülich,
NIC Series, Vol. **42**, ISBN 978-3-9810843-8-2, pp. 507-534, 2009.

http://www.fz-juelich.de/nic-series/volume42

Wavelets and Their Application for the Solution of Poisson's and Schrödinger's Equation

Stefan Goedecker

Department of Physics
University of Basel, Switzerland
E-mail: stefan.goedecker@unibas.ch

Wavelets can be used as a basis set for the solution of partial differential equations. After introducing the theoretical framework of wavelet theory, we will show how they can be used to solve Poisson's equation and Schrödinger's equation in an efficient way.

1 Wavelets, an Optimal Basis Set

The preferred way to solve partial differential equations is to express the solution as a linear combination of so-called basis functions. These basis functions can for instance be plane waves, Gaussians or finite elements. Having discretized the differential equation in this way makes it amenable to a numerical solution. In the case of Poisson's equation one obtains for instance a linear system of equation, in the case of Schrödinger's equation one obtains an eigenvalue problem. This procedure is usually more stable than other methods which do not involve basis functions, such as finite difference methods. Wavelets[3,2] are just another basis set which however offers considerable advantages over alternative basis sets and allows us to attack problems not accessible with conventional numerical methods. Its main advantages are:

- The basis set can be improved in an systematic way:
  If one wants the solution of the differential equation with higher accuracy one can just add more wavelets in the expansion of the solution. This will not lead to any numerical instabilities as one encounters for instance with Gaussians. The accuracy of the solution is determined by one single parameter similar to the minimal wavelength determining the accuracy of a plane wave expansion. In the case of the Gaussian type basis sets used in quantum chemistry there are many parameters which determine the accuracy and it is frequently not obvious which one has the largest leverage to improve upon the accuracy.

- Different resolutions can be used in different regions of space:
  If the solution of the differential equation is varying particularly rapidly in a particular region of space one can increase the resolution in this region by adding more high resolution wavelets centered around this region. This varying resolution is for instance not possible with plane waves, which give the same resolution in the whole computational volume.

- The coupling between different resolution levels is easy:
  Finite elements can also be used with varying resolution levels. The resulting highly structured grids lead however to very complicated matrix structures, requiring indirect

indexing of most arrays. In the case of wavelets, in contrast, the coupling between different resolution levels is rather easy.

- There are few topological constraints for increased resolution regions:
  The regions of increased resolution can be chosen arbitrarily, the only requirement being that a region of higher resolution be contained in a region of the next lower resolution. If one uses for instance generalized plane waves in connection with curvilinear coordinates[1] to obtain varying resolution one has the requirement that the varying resolution grid can be obtained by a mapping from a equally spaced grid. Increasing the resolution in one region requires decreasing the resolution in some other region.

- The Laplace operator is diagonally dominant in an appropriate wavelet basis:
  This allows for a simple but efficient preconditioning scheme for equations such as Poisson's or Schrödinger's equation which contains the Laplacian as well. As a result the number of iterations needed in the iterative solution of the linear algebra equations corresponding to these differential equations is fairly small and independent of the maximal resolution. No such easy and efficient preconditioning scheme is known for other varying resolution schemes such as finite elements, Gaussians or generalized plane waves with curvilinear coordinates.

- The matrix elements of the Laplace operator are very easy to calculate:
  The requirement that the matrix elements can easily be calculated is essential for any basis set and therefore fulfilled by all standard basis sets. For the case of wavelets it is however particularly easy since they can be calculated on the fly by simple scaling arguments and therefore need not be stored in memory.

- The numerical effort scales linearly with respect to system size:
  Three-dimensional problems of realistic size require usually a very large number of basis functions. It is therefore of utmost importance, that the numerical effort scales only linearly (and not quadratically or cubically) with respect to the number of basis functions. If one uses iterative matrix techniques, this linear scaling can only be obtained if two requirements are fulfilled, namely that the matrix vector multiplications which are necessary for all iterative methods can be done with linear scaling and that the number of matrix vector multiplications is independent of the problem size. The first requirement is fulfilled if either the matrix representing the differential operator is sparse or can be transformed into sparse form by a transformation which has linear scaling, a requirement fulfilled by wavelets. The second requirement is related to the availability of a good preconditioning scheme. Since such a scheme exists, the conditioning number of the involved matrices do not vary strongly with respect to the problem size and the number of iterations (i.e. matrix vector multiplications) is independent of the problem size.

2 A First Tour of Some Wavelet Families

Many families of wavelets have been proposed in the mathematical literature. If one wants to use wavelets for the solution of differential equations, one therefore has to choose one specific family which is most advantageous for the intended application. Within one family

there are also members of different degree. Without going into any detail at this point we will in the following just show some plots of some common wavelet families. Only families with compact support (i.e. they are nonzero only in a finite interval) will be presented. All these wavelet families can be classified as either being an orthogonal or biorthogonal family. The meaning of orthogonality will be explained later. Each orthogonal wavelet family is characterized by two functions, the mother scaling function ϕ and the mother wavelet ψ. In the case of biorthogonal families one has a dual scaling function $\tilde{\phi}$ and a dual wavelet $\tilde{\psi}$ in addition to the non-dual quantities.

Figure 1 shows the orthogonal Haar wavelet family, which is conceptually the simplest wavelet family. It is too crude to be useful for any numerical work, but its simplicity will help us to illustrate some basic wavelet concepts. The Haar wavelet is identical to the zero-th degree Daubechies[3] wavelet.

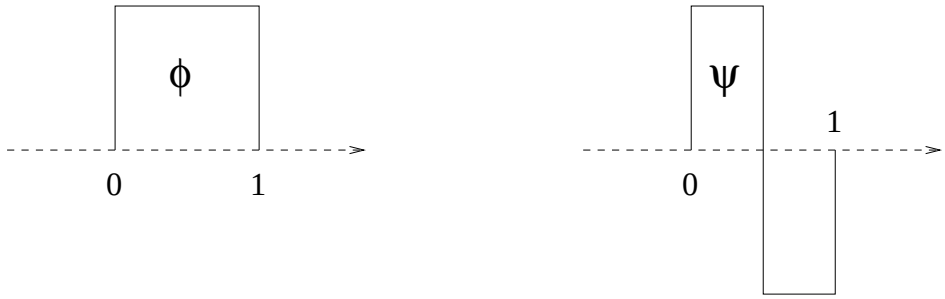


Figure 1. The Haar scaling function ϕ and wavelet ψ.

Figure 2 shows the 4 and 8 order Daubechies wavelets. Note that both the regularity and the support length increase with increasing order of the wavelets. The Daubechies family is an orthogonal wavelet family.

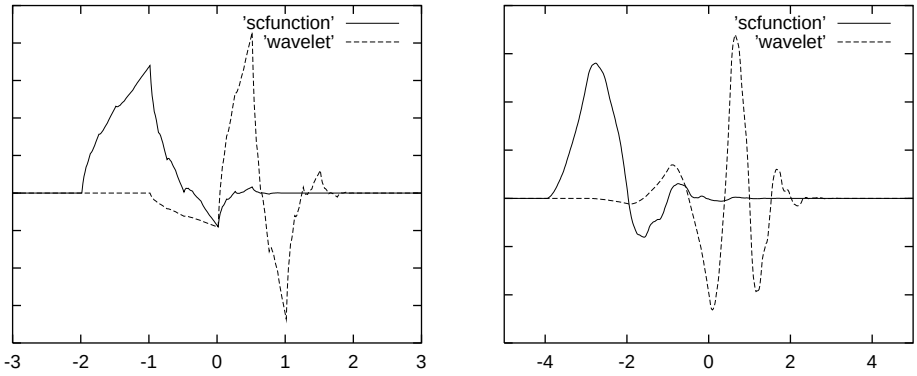


Figure 2. The orthogonal Daubechies scaling function and wavelet of degree 4 (left panel) and 8 (right panel).

Figure 3 shows a biorthogonal interpolating wavelet family of degree 4. It is smoother than other families of the same degree. Note that the scaling function vanishes at all integer points except at the origin.

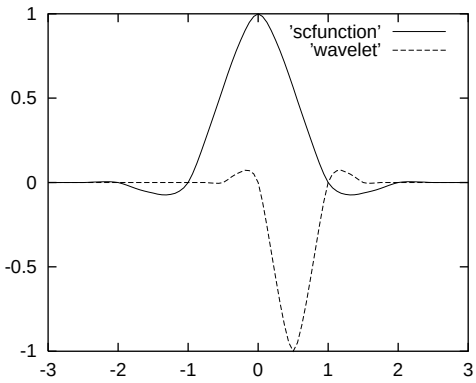


Figure 3. The interpolating scaling function and wavelet of degree 4.

3 Forming a Basis Set

To obtain a basis set at a certain resolution level k one can use all the integer translations of the mother scaling function of some wavelet family,

$$\phi_i^k(x) \propto \phi(2^k x - i)\,. \tag{1}$$

Note that with this convention higher resolution corresponds to larger values of k. Since high resolution scaling functions are skinnier, more translation indices i are allowed for a interval of fixed length. Some examples for an unspecified wavelet family are shown in Figure 4.

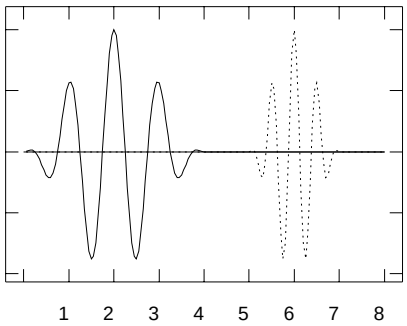


Figure 4. Two basis functions $\phi_2^0(x)$ (solid line) and $\phi_{12}^1(x)$ (dotted line) for an arbitrary wavelet family.

Exactly the same scaling and shifting operations can of course also be applied to the wavelets,

$$\psi_i^k(x) \propto \psi(2^k x - i)\,. \tag{2}$$

This set of wavelet basis functions can be added as a basis to the scaling functions as will be explained in the following.

4 The Haar Wavelet

In the case of the Haar family, any function which can exactly be represented at any level of resolution is necessarily piecewise constant. One such function is shown in Figure 5.

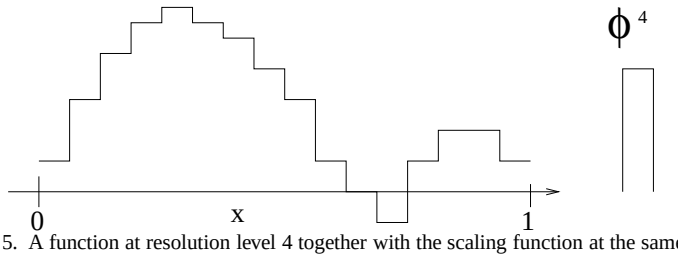


Figure 5. A function at resolution level 4 together with the scaling function at the same resolution level.

Evidently this function can be written as a linear combination of the scaling functions $\phi_i^4(x)$

$$f = \sum_{i=0}^{15} s_i^4\, \phi_i^4(x)\,, \tag{3}$$

where $s_i^4 = f(i/16)$.

Another, more interesting, possibility consists of expanding a function with respect to both scaling functions and wavelets of different resolution. Even though such a expansion contains both scaling functions and wavelets, we will refer to it as a wavelet representation to distinguish it from the our scaling function representation of Equation (3). A wavelet representation is possible because a scaling function at resolution level k is always a linear combination of a scaling function and a wavelet at the next coarser level $k-1$ as shown in Figure 6.

Using this relation depicted in Figure 6, we can write any linear combination of the two scaling functions $\phi_{2i}^k(x)$ and $\phi_{2i+1}^k(x)$ as a linear combination of $\phi_i^{k-1}(x)$ and $\psi_i^{k-1}(x)$. Hence we can write f as

$$f = \sum_{i=0}^{7} s_i^3\, \phi_i^3(x) + \sum_{i=0}^{7} d_i^3\, \psi_i^3(x)\,. \tag{4}$$

It is easy to verify that the transformation rule for the coefficients is given by

$$s_i^{k-1} = \frac{1}{2} s_{2i}^k + \frac{1}{2} s_{2i+1}^k \qquad ; \qquad d_i^{k-1} = \frac{1}{2} s_{2i}^k - \frac{1}{2} s_{2i+1}^k\,. \tag{5}$$

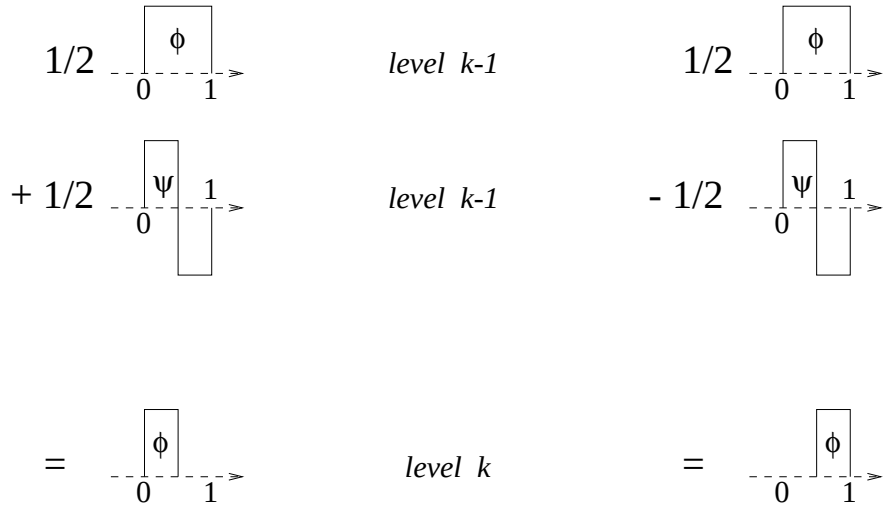


Figure 6. A skinny (level k) scaling function is a linear combination of a fat (level $k-1$) scaling function and a fat wavelet.

So to calculate the expansion coefficients with respect to the scaling functions at the next coarser level, we have to take an average over expansion coefficients at the finer resolution level. Because we have to take some weighted sum these coefficients are denoted by s. To get the expansion coefficients with respect to the wavelet, we have to take some weighted difference and the coefficients are accordingly denoted by d. The wavelet part contains mainly high frequency components and by doing this transformation we therefore peel off the highly oscillatory parts of the function. The remaining part represented by the coefficients s_i^{k-1} is therefore smoother. It is admittedly difficult to talk about smoothness for this kind of piecewise constant functions. This effect will be more visible for better wavelet families discussed later. For the case of our example in Figure 5 this remaining part after one transformation step is shown in Figure 7.

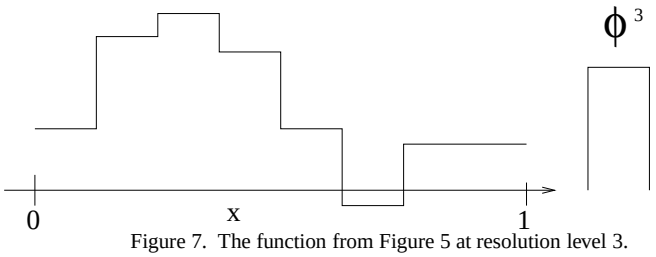


Figure 7. The function from Figure 5 at resolution level 3.

For any data set whose size is a power of 2, we can now apply this transformation repeatedly. In each step the number of s coefficients will be cut into half. So we have

to stop the procedure as soon as there is only one s coefficient left. Such a series of transformation steps is called a forward Haar wavelet transform. The resulting wavelet representation of the function in Equation (3) is then

$$f = s_0^0\phi_0^0(x) + d_0^0\psi_0^0(x) + \sum_{i=0}^{1} d_i^1\psi_i^1(x) + \sum_{i=0}^{3} d_i^2\psi_i^2(x) + \sum_{i=0}^{7} d_i^3\psi_i^3(x)\,. \qquad (6)$$

Note that in both cases we need exactly 16 coefficients to represent the function. In the coming sections such wavelet representations will be the focus of our interest.

By doing a backward wavelet transform, we can go back to the original scaling function representation of Equation (3). Starting at the coarsest resolution level, we have to express each scaling function and wavelet on the coarse level in terms of scaling functions at the finer level. This can be done exactly because wavelet families satisfy the so-called refinement relations depicted in Figure 8 for the Haar family.

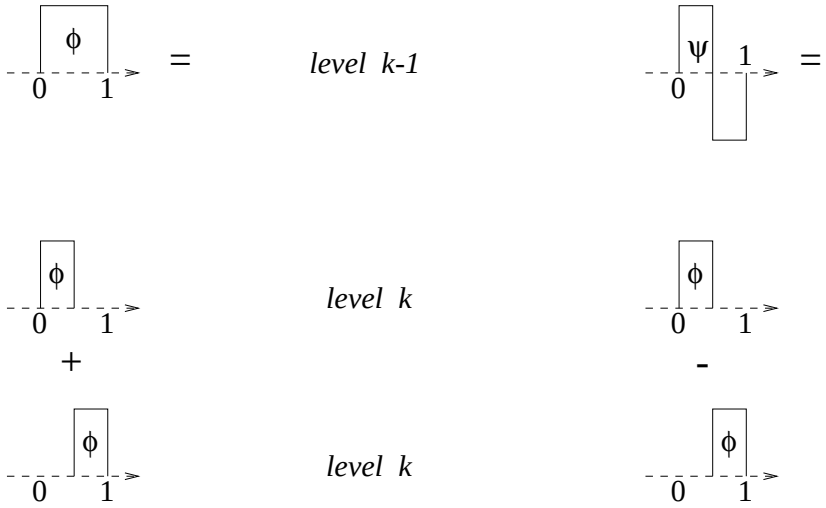


Figure 8. Fat (level $k-1$) scaling functions and fat wavelets are linear combinations of skinny (level k) scaling functions.

It then follows that we have to back-transform the coefficients in the following way

$$s_{2i}^{k+1} = s_i^k + d_i^k \qquad ; \qquad s_{2i+1}^{k+1} = s_i^k - d_i^k\,. \qquad (7)$$

5 The Concept of Multi-Resolution Analysis

In the previous sections a very intuitive introduction to wavelet theory was given. The formal theory behind wavelets is called Multi-Resolution Analysis[3] (MRA). Even though the formal definitions of MRA are usually not required for practical work, we will for completeness briefly present them. The equations which are useful for numerical work will be listed afterwards.

5.1 Formal definition of Multi-Resolution Analysis

- A Multi-Resolution Analysis consists of a sequence of successive approximation spaces V_k and associated dual spaces $\tilde{V}_k$, (which turn out to be the scaling function spaces and their dual counterpart) satisfying

$$V_k \subset V_{k+1} \qquad ; \qquad \tilde{V}_k \subset \tilde{V}_{k+1} \; .$$

- If a function $f(x)$ is contained in the space V_k, the compressed function $f(2x)$ has to be contained in the higher resolution space V_{k+1},

$$f(x) \in V_k \Leftrightarrow f(2x) \in V_{k+1} \qquad ; \qquad f(x) \in \tilde{V}_k \Leftrightarrow f(2x) \in \tilde{V}_{k+1} \; .$$

- If a function $f(x)$ is contained in the space V_k, its integer translate has to be contained in the same space,

$$f(x) \in V_0 \Leftrightarrow f(x+1) \in V_0 \qquad ; \qquad f(x) \in \tilde{V}_0 \Leftrightarrow f(x+1) \in \tilde{V}_0 \; .$$

- The union of all these spaces is the $L^2(\Re)$ space,

$$\overline{\bigcup_k V_k} = L^2(\Re) \; .$$

- There exists a biorthogonal pair of functions spanning V_k,

$$\int \tilde{\phi}_i^k(x) \, \phi_j^k(x) \, dx = \delta_{i,j} \; .$$

The wavelet spaces W_k, $\tilde{W}_k$ are then defined as the complement (orthogonal complement in the case of orthogonal families) of V_k in V_{k+1},

$$V_{k+1} = V_k \oplus W_k \qquad ; \qquad \tilde{V}_{k+1} = \tilde{V}_k \oplus W_k \; .$$

5.2 Basic formulas for biorthogonal wavelet families

The formal MRA requirements listed above lead to the following useful basic facts of wavelet analysis. The interested reader can find the nontrivial proofs of these formulas in the book by Daubechies[3].

- A biorthogonal wavelet family of degree m is characterized by 4 finite filters denoted by h_j, $\tilde{h}_j$, g_j, $\tilde{g}_j$. Since we will mainly deal with symmetric wavelet families, whose filters have a natural symmetry center, we will adopt a convention where the nonzero filter elements are in the interval $j = -m, ..., m$, and where m is even. In case the number of nonzero filter elements does not fit into this convention, it is always possible to pad the filters on both sides with zeroes, and to increase m artificially until it is compatible with this convention.

The filter coefficients satisfy the orthogonality relations

$$\sum_l h_{l-2i}\,\tilde{h}_{l-2j} = \delta_{i,j}\,, \tag{8}$$

$$\sum_l g_{l-2i}\,\tilde{g}_{l-2j} = \delta_{i,j}\,, \tag{9}$$

$$\sum_l h_{l-2i}\,\tilde{g}_{l-2j} = 0\,, \tag{10}$$

$$\sum_l \tilde{h}_{l-2i}\,g_{l-2j} = 0 \tag{11}$$

and the symmetry relations

$$g_{i+1} = (-1)^{i+1}\tilde{h}_{-i}\,, \tag{12}$$

$$\tilde{g}_{i+1} = (-1)^{i+1}h_{-i}\,. \tag{13}$$

- Scaling functions and wavelets at a coarse level can be written as the following linear combinations of scaling functions at a higher resolution level. These equations are called refinement relations,

$$\phi(x) = \sum_{j=-m}^{m} h_j\,\phi(2x-j)\,, \tag{14}$$

$$\psi(x) = \sum_{j=-m}^{m} g_j\,\phi(2x-j)\,, \tag{15}$$

$$\tilde{\phi}(x) = 2\sum_{j=-m}^{m} \tilde{h}_j\,\tilde{\phi}(2x-j)\,, \tag{16}$$

$$\tilde{\psi}(x) = 2\sum_{j=-m}^{m} \tilde{g}_j\,\tilde{\phi}(2x-j)\,. \tag{17}$$

In terms of the the two index multi level basis functions defined by,

$$\phi_i^k(x) = \phi(2^k x - i)\,, \tag{18}$$

$$\psi_i^k(x) = \psi(2^k x - i)\,, \tag{19}$$

$$\tilde{\phi}_i^k(x) = 2^k\tilde{\phi}(2^k x - i)\,, \tag{20}$$

$$\tilde{\psi}_i^k(x) = 2^k\tilde{\psi}(2^k x - i)\,, \tag{21}$$

the refinement relations are,

$$\phi_i^k(x) = \sum_{j=-m}^{m} h_j \, \phi_{2i+j}^{k+1}(x) \, , \tag{22}$$

$$\psi_i^k(x) = \sum_{j=-m}^{m} g_j \, \phi_{2i+j}^{k+1}(x) \, , \tag{23}$$

$$\tilde{\phi}_i^k(x) = \sum_{j=-m}^{m} \tilde{h}_j \, \tilde{\phi}_{2i+j}^{k+1}(x) \, , \tag{24}$$

$$\tilde{\psi}_i^k(x) = \sum_{j=-m}^{m} \tilde{g}_j \, \tilde{\phi}_{2i+j}^{k+1}(x) \, . \tag{25}$$

- A wavelet analysis (forward) transform is given by

$$s_i^{k-1} = \sum_{j=-m}^{m} \tilde{h}_j s_{j+2i}^k \, , \tag{26}$$

$$d_i^{k-1} = \sum_{j=-m}^{m} \tilde{g}_j s_{j+2i}^k \, .$$

A wavelet synthesis (backward) transform is given by

$$s_{2i}^{k+1} = \sum_{j=-m/2}^{m/2} h_{2j} \, s_{i-j}^k + g_{2j} \, d_{i-j}^k \tag{27}$$

$$s_{2i+1}^{k+1} = \sum_{j=-m/2}^{m/2} h_{2j+1} \, s_{i-j}^k + g_{2j+1} \, d_{i-j}^k \, .$$

These two equations are generalizations of equations (5), (7) that we derived in an intuitive way.

The wavelet transform is in principle for periodic data sets. Therefore the subscripts of the s and d coefficients have to be wrapped around once they are out of bounds.

- The fundamental functions satisfy the following orthogonality relations,

$$\int \tilde{\phi}_i^k(x) \phi_j^k(x) dx = \delta_{i,j} \, , \tag{28}$$

$$\int \tilde{\psi}_i^k(x) \phi_j^q(x) dx = 0 \, , \; k \geq q \, , \tag{29}$$

$$\int \psi_i^k(x) \tilde{\phi}_j^q(x) dx = 0 \, , \; k \geq q \, , \tag{30}$$

$$\int \psi_i^k(x) \tilde{\psi}_j^q(x) dx = \delta_{k,q} \delta_{i,j} \, . \tag{31}$$

5.3 Basic formulas for orthogonal wavelet families

- An orthogonal wavelet family of degree m is characterized by 2 finite filters denoted by h_j, g_j, satisfying the orthogonality relations

$$\sum_l h_{l-2i}\, h_{l-2j} = \delta_{i,j}\ , \tag{32}$$

$$\sum_l g_{l-2i}\, g_{l-2j} = \delta_{i,j}\ , \tag{33}$$

$$\sum_l h_{l-2i}\, g_{l-2j} = 0 \tag{34}$$

  and the symmetry relation

$$g_{i+1} = (-1)^{i+1} h_{-i}\ . \tag{35}$$

- The refinement relations are

$$\phi(x) = \sqrt{2} \sum_{j=-m}^{m} h_j\, \phi(2x-j)\ , \tag{36}$$

$$\psi(x) = \sqrt{2} \sum_{j=-m}^{m} g_j\, \phi(2x-j)\ . \tag{37}$$

  In terms of the the two index multi level basis functions defined by

$$\phi_i^k(x) = \sqrt{2}^k \phi(2^k x - i)\ , \tag{38}$$

$$\psi_i^k(x) = \sqrt{2}^k \psi(2^k x - i)\ , \tag{39}$$

  the refinement relations are

$$\phi_i^k(x) = \sum_{j=-m}^{m} h_j\, \phi_{2i+j}^{k+1}(x)\ , \tag{40}$$

$$\psi_i^k(x) = \sum_{j=-m}^{m} g_j\, \phi_{2i+j}^{k+1}(x)\ . \tag{41}$$

- The formulas for the forward and backward wavelet transforms are identical to the biorthogonal case (Equation (26) and (27)), with the exception that the filters $\tilde{h}$ and $\tilde{g}$ have to be replaced by the filters h and g in the forward transform.

- The fundamental functions satisfy the orthogonality relations,

$$\int \phi_i^k(x)\phi_j^k(x)dx = \delta_{i,j}\ , \tag{42}$$

$$\int \psi_i^k(x)\phi_j^q(x)dx = 0\ ,\ k \geq q\ , \tag{43}$$

$$\int \psi_i^k(x)\psi_j^q(x)dx = \delta_{k,q}\delta_{i,j}\ . \tag{44}$$

6 The Fast Wavelet Transform

One single sweep in a wavelet transformation (Eq. 26, Eq. 27) is a convolution with a short filter that can be done with linear scaling with respect to the size of the data set. An entire wavelet analysis transformation consists of several sweeps where in each consecutive sweep the amount of data to be transformed is cut into half. The total number of arithmetic operations is therefore given by a geometric series and is proportional to the data set. More precisely, if our filters h and g have length $2m$ the operation count is given by $2m(n+n/2+n/4+...) < 4mn$. The entire wavelet analysis scales therefore linearly. An entire wavelet synthesis is just the reverse operation and scales linearly as well. Below the evolution of a data set in a wavelet analysis is shown.

original data

$$s_0^4\ s_1^4\ s_2^4\ s_3^4\ s_4^4\ s_5^4\ s_6^4\ s_7^4\ s_8^4\ s_9^4\ s_{10}^4\ s_{11}^4\ s_{12}^4\ s_{13}^4\ s_{14}^4\ s_{15}^4\ =\ S^4$$

after first sweep

$$s_0^3\ s_1^3\ s_2^3\ s_3^3\ s_4^3\ s_5^3\ s_6^3\ s_7^3\ d_0^3\ d_1^3\ d_2^3\ d_3^3\ d_4^3\ d_5^3\ d_6^3\ d_7^3\ =\ S^3, D^3$$

after second sweep

$$s_0^2\ s_1^2\ s_2^2\ s_3^2\ d_0^2\ d_1^2\ d_2^2\ d_3^2\ d_0^3\ d_1^3\ d_2^3\ d_3^3\ d_4^3\ d_5^3\ d_6^3\ d_7^3\ =\ S^2, D^2, D^3$$

after third sweep

$$s_0^1\ s_1^1\ d_0^1\ d_1^1\ d_0^2\ d_1^2\ d_2^2\ d_3^2\ d_0^3\ d_1^3\ d_2^3\ d_3^3\ d_4^3\ d_5^3\ d_6^3\ d_7^3\ =\ S^1, D^1, D^2, D^3$$

final data

$$s_0^0\ d_0^0\ d_0^1\ d_1^1\ d_0^2\ d_1^2\ d_2^2\ d_3^2\ d_0^3\ d_1^3\ d_2^3\ d_3^3\ d_4^3\ d_5^3\ d_6^3\ d_7^3\ =\ S^0, D^0, D^1, D^2, D^3$$

Note that this transformation from the original data to the final data corresponds exactly to the transformation done in an intuitive way to get from Equation (3) to Equation (6).

7 Interpolating Wavelets

As will be discussed later, interpolating wavelets have many properties, which make them highly suitable as basis sets for partial differential equations. At the same time they are conceptionally the simplest wavelets. We will therefore describe the construction of the elementary interpolating wavelet[7,6] in detail.

The construction of interpolating wavelets is closely connected to the question of how to construct a continuous function $f(x)$ if only its values f_i on a finite number of grid points i are known. One way to do this is by recursive interpolation. In a first step we interpolate the functional values on all the midpoints by using for instance the functional values of two grid points to the right and of two grid points to the left of the midpoint. These four functional values allow us to construct a third order polynomial and we can then evaluate it at the midpoint. In the next step, we take this new data set, which is now twice as large as the original one, as the input for a new midpoint interpolation procedure. This can be done recursively ad infinitum until we have a quasi continuous function.

Let us now show, how this interpolation prescription leads to a set of basis functions. Denoting by the Kronecker δ_{i-j} a data set whose elements are all zero with the exception of the element at position j, we can write any initial data set as a linear combination of such Kronecker data sets: $f_i = \sum_j f_j \delta_{i-j}$. Now the whole interpolation procedure is clearly linear, i.e. the sum of two interpolated values of two separate data sets is equal to the interpolated value of the sum of these two data sets. This means that we can instead also take all the Kronecker data sets as the input for separate ad-infinitum interpolation procedures, to obtain a set of functions $\phi(x - j)$. The final interpolated function is then identical to $f(x) = \sum_j f_j \phi(x - j)$. If the initial grid values f_i were the functional values of a polynomial of degree less than four, we obviously will have exactly reconstructed the original function from its values on the grid points. Since any smooth function can locally be well approximated by a polynomial, these functions $\phi(x)$ are good basis functions and we will use them as scaling functions to construct a wavelet family.

The first construction steps of an interpolating scaling function are shown in Figure 9 for the case of linear interpolation. The initial Kronecker data set is denoted by the big dots. The additional data points obtained after the first interpolation step are denoted by medium size dots and the additional data points obtained after the second step by small dots.

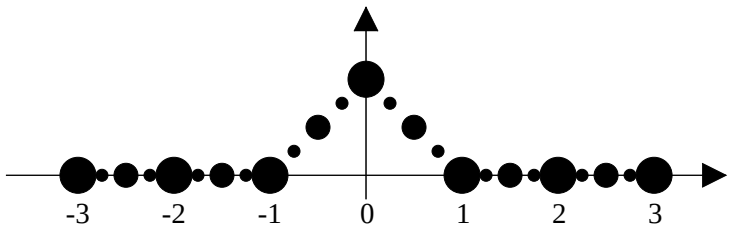


Figure 9. The first two steps of a recursive interpolation procedure in the case of simple linear interpolation. The original data point are represented by the big dots, data points filled in by the following two interpolation steps by medium and small dots.

Continuing this process ad infinitum will then result in the function shown in the left panel of Figure 10. If an higher order interpolation scheme is used the function shown in the right panel of Figure 10 is obtained.
By construction it is clear, that $\phi(x)$ has compact support. If an $(m-1)$-th order interpolation scheme is used, the filter length is $(m-1)$ and the support interval of the scaling function is $[-(m-1); (m-1)]$.

It is also not difficult to see that the function $\phi(x)$ satisfies the refinement relation. Let us again consider the interpolation ad infinitum of a Kronecker data set which has everywhere zero entries except at the origin. We can now split up this process into the first step where we calculate the half-integer grid point values and a remaining series of separate ad infinitum interpolations for all half-integer Kronecker data sets, which are necessary to represent the data set obtained by the first step. Doing the ad-infinitum interpolation for a half integer data set with a unit entry at (half integer) position j, we obviously obtain the same scaling function, just compressed by a factor of 2, $\phi(2x - j)$. If we are using a $(m-1)$-th order interpolation scheme (i.e. m input data for the interpolation process) we

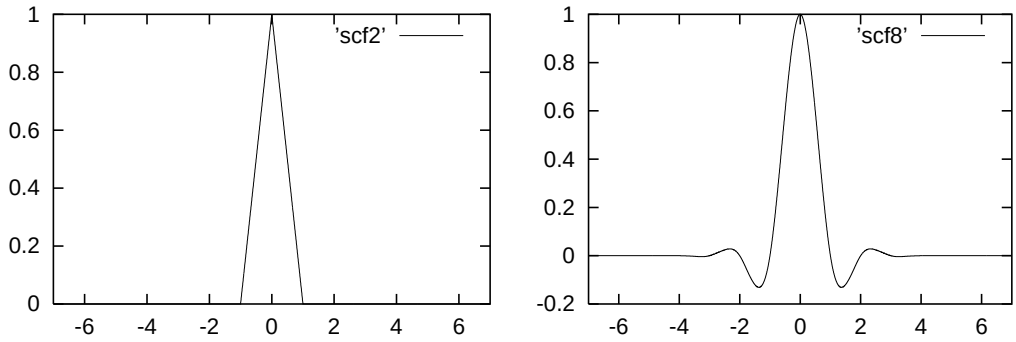


Figure 10. A Kronecker delta interpolated ad infinitum with linear interpolation (left panel) and 7-th order interpolation (right panel) .

thus get the relation

$$\phi(x) = \sum_{j=-m+1}^{m-1} \phi(j/2)\ \phi(2x-j)\ . \tag{45}$$

Comparing this equation with the refinement relation Equation (14) we can identify the first filter h as

$$h_j = \phi(j/2)\ ,\ \ j = -m+1, m-1\ .$$

For the case of third order interpolation the numerical values of h follow from the standard interpolation formula and are given by:

```
h =   {-1/16 , 0 , 9/16 , 1 , 9/16 , 0 , -1/16}
j =      -3    -2    -1    0     1    2     3
```

Let us next determine the filter $\tilde{h}$. Let us consider a function $f(x)$ which is band-limited in the wavelet sense, i.e which can exactly be represented by a superposition of scaling functions at a certain resolution level K

$$f(x) = \sum_j c_j \phi_j^K(x)\ .$$

It then follows from the orthogonality relation Equation (28) that

$$c_j = \int \tilde{\phi}_j^K(x) f(x) dx\ .$$

Now we have seen above that with respect to interpolating scaling functions, a band-limited function is just any polynomial of degree less than or equal to $m-1$, and that in this case the expansion coefficients c_j are just the functional values at the grid points (Equation (45)). We therefore have

$$\int \tilde{\phi}_j^K(x) f(x) dx = f_j\ ,$$

which shows that the dual scaling function $\tilde{\phi}$ is the delta function. Obviously the delta function satisfies a trivial refinement relation $\delta(x) = 2\delta(2x)$ and from Equation (16) we conclude that $\tilde{h}_j = \delta_j$.

```
ht =   {  0 ,  0  , 1 ,  0  , 0  }
j  =      -2   -1    0    1    2
```

From the symmetry Equations (12), (13) for the filters we can now determine the two remaining filters and we have thus completely specified our wavelet family. For $\tilde{g}_j$ we obtain

```
gt =   {  0 , 0 , -1/16 , 0 , 9/16 , -1 , 9/16 , 0 , -1/16 }
j  =      -4  -3    -2    -1    0      1    2     3     4
```

For g_j we obtain

```
g =   {  0 ,  0 ,  0 , -1 ,  0 }
j =      -2   -1    0    1    2
```

As expected, these 4 filters satisfy the orthogonality conditions (8) to (11).

Due to the easy structure of the filters in this case, the backward transform can be done by inspection to obtain the form of the 4 fundamental functions ϕ, ψ, $\tilde{\phi}$ and $\tilde{\psi}$. In the case of the scaling function ϕ, the d coefficients at all resolution levels vanish. For the even elements Equation (27) becomes

$$s_{2i}^{k+1} = s_i^k \; .$$

So the even grid points on the finer grid take on the values of the coarser grid. The odd filter elements are just the interpolating coefficients giving:

$$s_{2i+1}^{k+1} = h_3 \, s_{i-1}^k + h_1 \, s_{i+0}^k + h_{-1} \, s_{i+1}^k + h_{-3} \, s_{i+2}^k \; .$$

So the values at the odd fine grid points are just interpolated from the coarse grid points. In summary we thus see that an infinite series of backward transforms just describes the ad-infinitum interpolation process depicted in Figure 9.

In the case of the wavelet ψ the only nonzero d coefficient in the input data will generate in the first step a s data set where again only one coefficient is nonzero, since the g filter has only one nonzero entry. Continuing the procedure one will thus obtain for the wavelet a negative scaling function compressed by a factor of 2, $\psi(x) = -\phi(2x-1)$.

To generate the dual functions $\tilde{\phi}$ and $\tilde{\psi}$, one has to replace the filters h and g in the backward transform by the dual counterparts $\tilde{h}$ and $\tilde{g}$. For the case of the dual scaling function $\tilde{\phi}$, one sees by inspection that the backward transform equations Equation (27) become:

$$s_{2i+1}^{k-1} = 0 \qquad ; \qquad s_{2i}^{k-1} = \begin{cases} 1 \text{ if } i = 0 \\ 0 \text{ otherwise} \end{cases}$$

As one should, one thus obtains a delta function

$$\tilde{\phi}(x) = \delta(x) \; . \tag{46}$$

For the case of a dual wavelet $\tilde{\psi}$ the argument is analogous to the non-dual case. In the first step of the backward transform the filter $\tilde{g}$ generates 5 nonzero s coefficients, which will become 5 delta functions through the action of the filter $\tilde{h}$. We get

$$\begin{aligned} \tilde{\psi}(x) = & -\tfrac{1}{16}\delta((x-\tfrac{1}{2})+3/2) + \tfrac{9}{16}\delta((x-\tfrac{1}{2})+1/2) - \delta((x-\tfrac{1}{2})) \\ & + \tfrac{9}{16}\delta((x-\tfrac{1}{2})+1/2) - \tfrac{1}{16}\delta((x-\tfrac{1}{2})+3/2) \; . \end{aligned} \tag{47}$$

We thus see that the interpolating wavelet is a very special case in that its scaling function and wavelet have the same functional form and that the dual functions are related to the delta function. The non-dual functions are shown in Figure 3 . Filters for interpolating wavelets of other degrees are given in the Appendix.

8 Expanding Polynomials in a Wavelet Basis

Functions of practical interest are of course not simple polynomials, and it will be discussed later how to expand arbitrary functions in a wavelet basis. For several proofs the expansion of polynomials in a wavelet basis is however important and we will therefore derive the following theorem: The scaling function expansion coefficients $s_i(l)$ of a polynomial of degree l are themselves a polynomial of the same degree l.

Let us first demonstrate the theorem for the trivial case of a constant, i.e. $l = 0$. The expansion coefficients $s_i(0)$ are given by $\int \tilde{\phi}(x-i)dx$. Assuming $\int \tilde{\phi}(x)dx$ is normalized to 1 we thus obtain $s_i(0) = 1$.

In the linear case (i.e. $l = 1$) we have $s_i(1) = \int \tilde{\phi}(x-i)xdx$. For the shifted coefficient we get

$$\begin{aligned} s_{i+1}(1) &= \int \tilde{\phi}(x-i-1)xdx = \int \tilde{\phi}(x-i)(x+1)dx \\ &= s_i(1) + 1 \; . \end{aligned} \tag{48}$$

So we see that $s_i(1)$ satisfies the difference equation for a linear polynomial and that it is therefore a linear polynomial.

For arbitrary degree l we get

$$\begin{aligned} s_{i+1}(l) &= \int \tilde{\phi}(x-i-1)x^l dx = \int \tilde{\phi}(x-i)(x+1)^l dx \\ &= \sum_\tau \int \tilde{\phi}(x-i)\frac{l!}{\tau!(l-\tau)!}x^\tau dx \\ &= \sum_\tau \frac{l!}{\tau!(l-\tau)!}s_i(\tau) \; . \end{aligned} \tag{49}$$

So wee see indeed that $s_i(l)$ is a polynomial of lth degree since it satisfies the corresponding difference equation, which proves the theorem.

9 Orthogonal Versus Biorthogonal Wavelets

The interpolating wavelets constructed above are a special case of so-called biorthogonal wavelet families. The interpolating wavelets have the property of being the smoothest ones for a fixed filter length. On the other hand the dual functions of the interpolating wavelet family are the least smooth ones. Loosely speaking the sum of the smoothness of the dual and non-dual space are a constant for a given filter length. For a given filter length one can therefore either go for maximum smoothness in the dual or non-dual space. The interpolating wavelets are your favorite choice if you want maximum smoothness in the non-dual space.

Wavelets are called orthogonal if the dual quantities are equal to the non-dual quantities. In the case of orthogonal wavelets the smoothness in dual and non-dual space is thus obviously the same. They are therefore not as smooth as the interpolating wavelets. The smoothness properties of the Daubechies family are actually not as bad as on might expect from looking at the "ugly" plots. With the 4-th order family one can exactly represent linear function, with the 6-th order family quadratic and with the 8-th order family cubic polynomials.

10 Expanding Functions in a Wavelet Basis

As we have seen, there are two possible representations of a function within the framework of wavelet theory. The first one is called scaling function representation and involves only scaling functions. The second is called wavelet representation and involves wavelets as well as scaling functions. Both representations are completely equivalent and exactly the same number of coefficients are needed. The scaling function representation is given by

$$f(x) = \sum_j s_j^{Kmax} \phi_j^{Kmax}(x) \; . \tag{50}$$

Evidently this approximation is more accurate if we use skinnier scaling functions from a higher resolution level $Kmax$. From the orthogonality relations (28) it follows, that the coefficients are given by

$$s_j^{Kmax} = \int \tilde{\phi}_j^{Kmax}(x) \; f(x) dx \; . \tag{51}$$

Once we have a set of coefficients s_j^{Kmax} we can use a full forward wavelet transform to obtain the wavelet representation

$$f(x) = \sum_j s_j^{Kmin} \; \phi_j^{Kmin}(x) + \sum_{K=Kmin}^{Kmax} \sum_j d_j^K \; \psi_j^K(x) \; . \tag{52}$$

Alternatively, one could also directly calculate the d coefficients by integration

$$d_j^K = \int \tilde{\psi}_j^K(x) \; f(x) dx \; . \tag{53}$$

The above Equation (53) follows from the orthogonality relations (29) to (31). So we see that if we want to expand a function either in scaling functions or wavelets, we have to perform integrations at some point to calculate the coefficients. For general wavelet families this integration is fairly cumbersome[5] and requires especially in 2 and 3 dimensions a substantial number of integration points. Furthermore it is not obvious how to do the integration if the function is only given in tabulated form. If one wants to obtain the same number of coefficients as one has functional values, one could either first interpolate the function to obtain the necessary number of integration points, which will introduce additional approximations. If one does not generate additional integration points, then the number of coefficients will necessarily be less than the number of functional values and information is thus lost. The interpolating wavelets discussed above are the glorious exception. Since the dual scaling function is a delta function and since the dual wavelet is a sum of the delta functions, one or a few data points are sufficient to do the integration exactly. In the case of periodic data sets, the filters will wrap around for data points close enough to the boundary of the periodic volume. One will therefore get exactly the same number of coefficients as one has data points and one has an invertible one-to-one mapping between the functional values on the grid and the expansion coefficients.

Non-periodic data sets can also be handled. In this case we have to put the non-periodic data set into a larger periodic data set consisting of zeroes. This composite data set will then contain the nonzero non-periodic data set in the middle surrounded by a layer of zeroes on all sides. If this layer of zeroes is broader than half the filter length $m/2$ opposite

ends will not interfere during one sweep of a wavelet transform and one obtains the correct representation of the non-periodic function. Correct means in this context that the value of the nonzero coefficients would not change if we made the surrounding layer of zeroes broader.

The interpolating wavelets are also unbeatable from the point of view of accuracy. The accuracy of a scaling function expansion depends on the smoothness of the scaling function. This is easy to see for the case of interpolating wavelets. The functional value of a scaling function expansion at any midpoint is given by interpolation and thus the error is also given by the well known interpolation error formula. If h is the grid spacing and $(m-1)$-th order interpolation is used then the error is proportional to h^m. So since the interpolating wavelets are the smoothest wavelets they are also the most accurate ones. If on the other hand one is willing to accept a certain error then the interpolating wavelets will meet this error criteria with the smallest number of expansion coefficients.

This fact can also be understood from a different point of view. Let us introduce the moments $\tilde{M}_l$,

$$\tilde{M}_l = \int \tilde{\psi}(x)\, x^l dx \; .$$

Now we know, that locally any smooth function can be approximated by a polynomial. Let us for simplicity consider the coefficient d_0^K at the origin,

$$d_0^K = \sum_{\nu=0}^{\infty} \int f^\nu(0) \frac{x^\nu}{\nu!}\, \tilde{\psi}_0^K(x) dx \; .$$

If the first L moments $l = 0, ..., L-1$ vanish this becomes

$$d_0^K = \sum_{\nu=L}^{\infty} h^\nu C_\nu \; ,$$

where we have used the fact that $\tilde{\psi}$ is a sum of delta functions and where C_ν are appropriate constants. The d coefficients decay therefore as h^L and since the error is proportional to the coefficients of the wavelets which are discarded, the error is proportional to h^L as well. In the case of the 4-th order interpolating wavelet it is easy to see, that the first 4 moments vanish, $\tilde{M}_l = 0, l = 0, 1, 2, 3$ and thus the error is indeed proportional to h^4. The measured decay of the d coefficients for the case of a Gaussian is shown in Figure 11.

This relation between the error in the expansion of a function and the number of vanishing moments is not only valid for interpolating wavelets but also holds true for other wavelet families.

11 Wavelets in 2 and 3 Dimensions

The easiest way to construct a wavelet basis in higher dimensional spaces is by forming product functions[3]. For simplicity of notation we will concentrate on the 2-dimensional case, the generalization to higher dimensional spaces being obvious. The space of all scaling functions of resolution level k is given in the 2-dimensional case by

$$\phi_{i1,i2}^k(x,y) = \phi_{i1}^k(x)\phi_{i2}^k(y) \; . \tag{54}$$

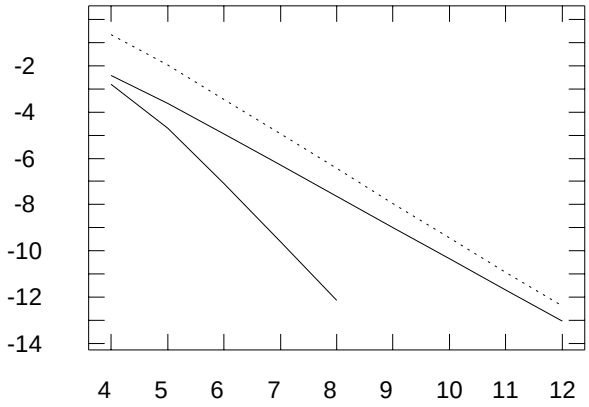


Figure 11. The decay of the d coefficients as a function of their resolution level on a double logarithmic scale. The solid lines show the result for a 4-th and 8-th order interpolating wavelet, the dashed line is for the 8-th order Daubechies family.

The wavelets consist of three types of products

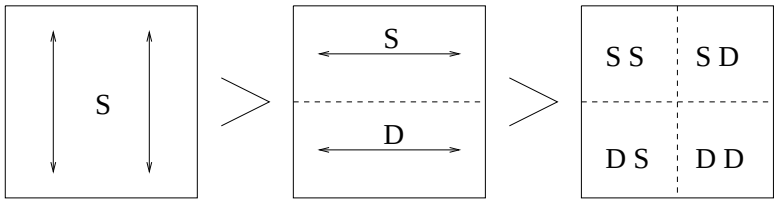
$$\psi[sd]^k_{i1,i2}(x,y) = \phi^k_{i1}(x)\psi^k_{i2}(y)\,, \tag{55}$$

$$\psi[ds]^k_{i1,i2}(x,y) = \psi^k_{i1}(x)\phi^k_{i2}(y)\,, \tag{56}$$

$$\psi[dd]^k_{i1,i2}(x,y) = \psi^k_{i1}(x)\psi^k_{i2}(y)\,. \tag{57}$$

In a 3-dimensional space the scaling functions are correspondingly of $[sss]$ type and there are 7 different classes of wavelets denoted by $[ssd]$, $[sds]$, $[sdd]$, $[dss]$, $[dsd]$, $[dds]$ and $[ddd]$.

It is easy to see, that both the many-dimensional scaling functions and wavelets satisfy refinement and orthogonality relations that are obvious generalizations of the 1-dimensional case. A wavelet transform step in the 2-dimensional setting is done by first transforming along the x and then along the y direction (or vice versa) as shown below.



To do a full 2-dim wavelet analysis one has to do a series of analysis steps. In each step the size of the active data set is reduced by 1/4 as shown in Figure 12. The total numerical effort therefore scales again linearly as in the one-dimensional case.

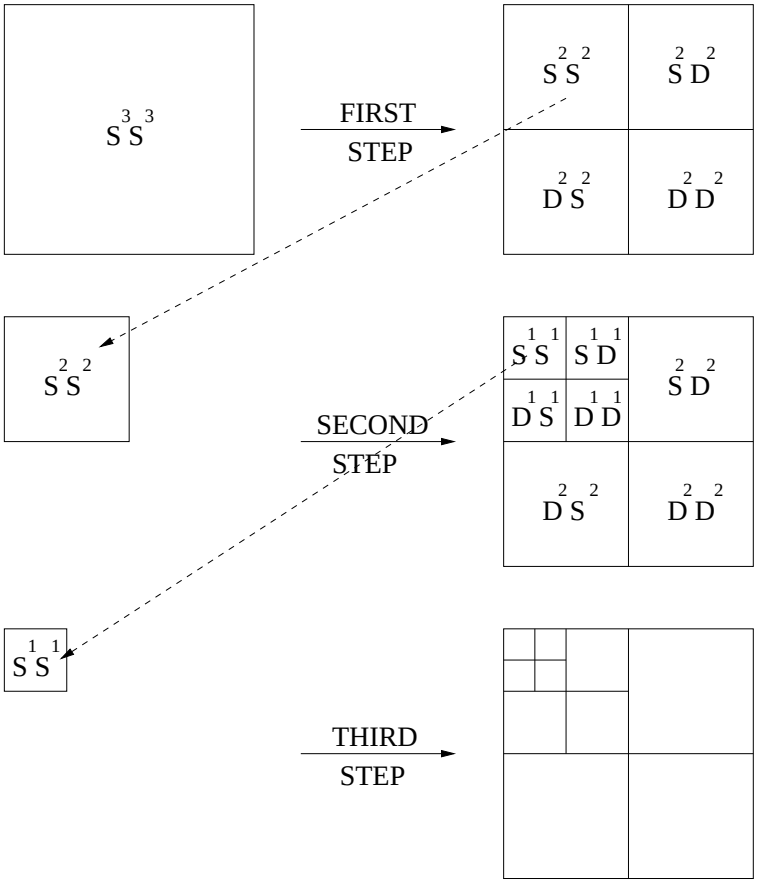


Figure 12. A full 2-dim wavelet analysis transformation.

12 Calculation of Differential Operators

As we have seen in the preceding chapter we need the matrix elements

$$\int \tilde{\phi}_i^k(x) \frac{\partial^l}{\partial x^l} \phi_j^k(x)\, dx\,, \tag{58}$$

$$\int \tilde{\psi}_i^k(x) \frac{\partial^l}{\partial x^l} \phi_j^k(x)\, dx\,, \tag{59}$$

$$\int \tilde{\phi}_i^k(x) \frac{\partial^l}{\partial x^l} \psi_j^k(x)\, dx\,, \tag{60}$$

$$\int \tilde{\psi}_i^k(x) \frac{\partial^l}{\partial x^l} \psi_j^k(x)\, dx\,, \tag{61}$$

to set up the SS, DS, SD and DD parts of the non-standard operator form. Matrix elements on different resolution levels k are obviously related by simple scaling relations. For instance

$$\int \tilde{\phi}_i^{k+1}(x)\frac{\partial^l}{\partial x^l}\phi_j^{k+1}(x)dx = 2^l \int \tilde{\phi}_i^k(x)\frac{\partial^l}{\partial x^l}\phi_j^k(x)dx \; . \tag{62}$$

So we just have to calculate these 4 matrix elements for one resolution level. On a certain resolution level, we can use the refinement relations to express the matrix elements involving wavelets in terms of matrix elements involving scaling functions (at a higher resolution level) only. Denoting the basic integral by a_i, where

$$a_i = \int \tilde{\phi}(x)\frac{\partial^l}{\partial x^l}\phi(x-i)dx \; , \tag{63}$$

we obtain

$$\int \tilde{\phi}_i(x)\frac{\partial^l}{\partial x^l}\phi_j(x)dx = a_{j-i} \; , \tag{64}$$

$$\int \tilde{\psi}_i(x)\frac{\partial^l}{\partial x^l}\phi_j(x)dx = 2^l \sum_{\nu,\mu} \tilde{g}_\nu h_\mu a_{2j-2i+\mu-\nu} \; , \tag{65}$$

$$\int \tilde{\phi}_i(x)\frac{\partial^l}{\partial x^l}\psi_j(x)dx = 2^l \sum_{\nu,\mu} \tilde{h}_\nu g_\mu a_{2j-2i+\mu-\nu} \; , \tag{66}$$

$$\int \tilde{\psi}_i(x)\frac{\partial^l}{\partial x^l}\psi_j(x)dx = 2^l \sum_{\nu,\mu} \tilde{g}_\nu g_\mu a_{2j-2i+\mu-\nu} \; . \tag{67}$$

To calculate a_i we follow Beylkin[9]. Using the refinement relations Equations (14) and (16) for ϕ and $\tilde{\phi}$ we obtain

$$\begin{aligned} a_i &= \int \tilde{\phi}(x)\frac{\partial^l}{\partial x^l}\phi(x-i)dx \qquad (68)\\ &= \sum_{\nu,\mu} 2\tilde{h}_\nu h_\mu \int \tilde{\phi}(2x-\nu)\frac{\partial^l}{\partial x^l}\phi(2x-2i-\mu)dx \\ &= \sum_{\nu,\mu} 2\tilde{h}_\nu h_\mu 2^{l-1} \int \tilde{\phi}(y-\nu)\frac{\partial^l}{\partial y^l}\phi(y-2i-\mu)dy \\ &= \sum_{\nu,\mu} \tilde{h}_\nu h_\mu 2^l \int \tilde{\phi}(y)\frac{\partial^l}{\partial y^l}\phi(y-2i-\mu+\nu)dy \\ &= \sum_{\nu,\mu} \tilde{h}_\nu h_\mu \; 2^l \; a_{2i-\nu+\mu} \end{aligned}$$

We thus have to find the eigenvector $\mathbf{a}$ associated with the eigenvalue of 2^{-l},

$$\sum_j A_{i,j} \; a_j = \left(\frac{1}{2}\right)^l a_i \; , \tag{69}$$

where the matrix $A_{i,j}$ is given by

$$A_{i,j} = \sum_{\nu,\mu} \tilde{h}_\nu h_\mu \, \delta_{j,2i-\nu+\mu} \; . \tag{70}$$

As it stands this eigensystem has a solution only if the rang of the matrix $A - 2^{-l} I$ is less than its dimension. For a well defined differential operator, i.e if l is less than the degree of smoothness of the scaling function this will be the case (for the 4-th order interpolating wavelet family the second derivative is for instance not defined). The system (69) is numerically unstable and it is therefore better to solve it using symbolic manipulations with a software such as Mathematica instead of solving it numerically.

The system of equations (69) determines the a_j's only up to a normalization factor. In the following, we will therefore derive the normalization equation. For simplicity, we will give the derivation only for the case of interpolating polynomials, even though the final result Equation (73) will hold in the general case.

From the normalization of the scaling function and from elementary calculus, it follows that

$$\int \phi(x) \frac{\partial^l}{\partial x^l} x^l \, dx = \int \phi(x) \; l! \, dx = l! \; . \tag{71}$$

On the other hand we know, that we can expand any polynomial of low enough degree exactly with the interpolating polynomials. The expansion coefficients are just i^l. So we obtain

$$\int \phi(x) \frac{\partial^l}{\partial x^l} \sum_i i^l \phi(x - i) = \sum_i i^l a_i \; . \tag{72}$$

By comparing Equation (71) and (72) we thus obtain the normalization condition

$$\sum_i i^l a_i = l! \; . \tag{73}$$

The interpolating wavelet family offers also an important advantage for the calculation of differential operators. Whereas in general derivative filters extend over the interval $[-2m; 2m]$ most of the border elements of interpolating wavelets are zero and their effective filter length is only $[-m + 2; m - 2]$.

Derivative filter coefficients for several families are listed in the Appendix.

13 Differential Operators in Higher Dimensions

As was pointed out before, higher dimensional wavelets can be constructed as products of one dimensional wavelets. The matrix elements of differential operators can therefore easily be derived.

Let us consider as an example the matrix elements of $\frac{\partial}{\partial x}$ with respect to the 2-dimensional scaling functions,

$$\int \tilde{\phi}^k_{i1,i2}(x,y) \frac{\partial}{\partial x} \phi^k_{j1,j2}(x,y) = \int \tilde{\phi}^k_{i1}(x) \tilde{\phi}^k_{i2}(y) \frac{\partial}{\partial x} \phi^k_{j1}(x) \phi^k_{j2}(y) = \delta_{i2-j2} a_{j1-i1} \; .$$

The remaining matrix elements among the wavelets and scaling functions can be derived along the same lines. Obviously a differential operator acting on x will only couple functions which have the same dependence with respect to y as indicated in Figure 13.

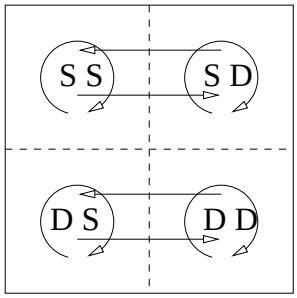


Figure 13. The coupling of the expansion coefficients under the action of a differential operator acting along the (horizontal) x axis.

14 The Solution of Poisson's Equation

In the following, a method[11] will be presented that uses interpolating scaling functions to solve Poisson's equation with free boundary conditions and $Nlog(N)$ scaling. The input is a charge density $\rho i1, i2, i3$ on a equally spaced 3-dimensional grid of $N = n_1 n_2 n_3$ grid points. For simplicity we put the grid spacing equal to 1. Since for interpolating scaling functions the expansion coefficients are just the values on the grid we can obtain from our discrete data set $\rho_{i1,i2,i3}$ a continuous charge distribution $\rho(\mathbf{r})$

$$\rho(\mathbf{r}) = \sum_{i1,i2,i3} \rho_{i1,i2,i3}\, \phi(x/h - i1)\, \phi(y/h - i2)\, \phi(z/h - i3) \tag{74}$$

It is not very difficult to prove that the discrete and continuous monopoles, i.e

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dz\ \rho(\mathbf{r}) = \sum_{i1,i2,i3} \rho_{i1,i2,i3}$$

In the same way the discrete and continuous dipoles are identical, i.e

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dz\ z\, \rho(\mathbf{r}) = \sum_{i1,i2,i3} i3\, \rho_{i1,i2,i3}$$

The potential on the grid point $j1, j2, j3$ (of same grid that was used for the input charge density) is then given by

$$\begin{aligned} V_{j1,j2,j3} &= \sum_{i1,i2,i3} \rho_{i1,i2,i3} \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dz \frac{\phi(x - i1)\, \phi(y - i2)\, \phi(z - i3)}{\sqrt{(x - j1)^2 + (y - j2)^2 + (z - j3)^2}} \\ &= \sum_{i1,i2,i3} \rho_{i1,i2,i3} F_{i1-j1,i2-j2,i3-j3} \end{aligned} \tag{75}$$

F is a long filter which depends only on the distance $i - j$ between the observation point j and the source point i. Since the above expression for the potential $V_{j1,j2,j3}$ is a convolution it can be calculated with FFT techniques at the cost of $N^3 \log(N^3)$ operations where N^3 is the number of grid points. It remains to calculate the values of the filter $F_{i1-j1,i2-j2,i3-j3}$. Calculating each of the N^3 filter elements as a 3-dimensional numerical integral would be too costly. The calculation becomes however feasible if the $1/r$

kernel is made separable. This can be achieved by representing it as a sum of Gaussians. The representation is best based on the identity

$$\frac{1}{r} = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-r^2 \exp(2s)+s} ds$$

Discretizing this integral we obtain

$$\frac{1}{r} = \sum_l w_l e^{-\gamma_l r^2} \tag{76}$$

With 89 well optimized values for w_l and γ_l it turns out that $1/r$ can be represented in the interval from 10^{-9} to 1 with an relative error of 10^{-8}. The 3-dimensional integral in Eq. 75 becomes then a sum of 89 terms each of which is a product of 1-dimensional integrals.

$$\int dx \int dy \int dz \frac{\phi(x-i1)\,\phi(y-i2)\,\phi(z-i3)}{\sqrt{(x-j1)^2+(y-j2)^2+(z-j3)^2}} =$$

$$\sum_{l=1}^{89} w_l \int dx \int dy \int dz\, \phi(x-i1)\,\phi(y-i2)\,\phi(z-i3)\, e^{-\gamma_l\left((x-j1)^2+(y-j2)^2+(z-j3)^2\right)} =$$

$$\sum_{l=1}^{89} w_l \int dx\, \phi(x-i1) e^{-\gamma_l (x-j1)^2} \int dy\, \phi(y-i2) e^{-\gamma_l (y-j2)^2} \int dz\, \phi(z-i3) e^{-\gamma_l (z-j3)^2}$$

Using 89 terms in Eq. 76 we have thus to solve just $89N$ one-dimensional integrals which can be done extremely rapidly on a modern computer. The main cost are thus the FFT's required to calculate the convolution with the kernel $F_{i1-j1,i2-j2,i3-j3}$.

The above presented method does not exploit the possibility to have adaptivity in a wavelet basis. Adpative methods to solve Poisson's equation on grids where the resolution varies by several orders of magnitude exist[10] as well. They are however based on more advanced concepts[4] such as non-standard operator forms and lifted interpolating wavelets.

15 The Solution of Schrödinger's Equation

Since the different Kohn-Sham orbitals in a density functional calculation have to be orthogonal, orthogonalization steps occur frequently in such a calculation. As a matter of fact these orthogonalization operations have cubic scaling and dominate thus the whole calculation for large system. It is therefore important that these operations can be done efficiently. This strongly suggests to use orthogonal Daubechies scaling functions and wavelets as basis functions for the Kohn-Sham orbitals. In spite of the striking advantages of Daubechies wavelets, the initial exploration of this basis set[8] did not lead to any algorithm that would be useful for real electronic structure calculations. This was due to the fact that an accurate evaluation of the local potential energy is difficult in a Daubechies wavelet basis. The kinetic energy part on the other hand is easy since it is just given by the Laplace operator. How to treat the Laplace operator has already been discussed. The obstacles in the evaluation of the potential energy have been overcome[12] recently and it was consequently shown that wavelets are an efficient basis set for electronic structure calculations[13] which outperforms plane waves for open structures. We will next discuss how

the different parts of the Hamiltonian are handled in a wavelet basis. For simplicity, we will discuss only the case where a wave function Ψ is expanded in scaling functions.

$$\Psi(\mathbf{r}) = \sum_{i_1,i_2,i_3} s_{i_1,i_2,i_3} \phi_{i_1,i_2,i_3}(\mathbf{r}) \tag{77}$$

The sum over i_1, i_2, i_3 runs over all the points of a uniform grid. The more general case of adaptive resolution is discussed in the original paper[13].

15.1 Treatment of local potential energy

The local potential $V(\mathbf{r})$ is generally known on the nodes of the uniform grid in the simulation box. Approximating a potential energy matrix element $V_{i,j,k;i',j',k'}$

$$V_{i,j,k;i',j',k'} = \int d\mathbf{r} \phi_{i',j',k'}(\mathbf{r}) V(\mathbf{r}) \phi_{i,j,k}(\mathbf{r})$$

by

$$V_{i,j,k;i',j',k'} \approx \sum_{l,m,n} \phi_{i,j,k}(\mathbf{r}_{l,m,n}) V(\mathbf{r}_{l,m,n}) \phi_{i,j,k}(\mathbf{r}_{l,m,n})$$

gives an extremely slow convergence rate with respect to the number of grid point used to approximate the integral because a single scaling function is not very smooth, i.e. it has a rather low number of continuous derivatives. A. Neelov and S. Goedecker[12] have shown that one should not try to approximate a single matrix element as accurately as possible but that one should try instead to approximate directly the expectation value of the local potential. The reason for this strategy is that the wave function expressed in the Daubechies basis is smoother than a single Daubechies basis function. A single Daubechies scaling function of order 16 has only 4 continuous derivatives. By suitable linear combinations of Daubechies 16 one can however exactly represent polynomials up to degree 7, i.e. functions that have 7 non-vanishing continuous derivatives. The discontinuities get thus canceled by taking suitable linear combinations. Since we use pseudopotentials, our exact wave functions are analytic and they can locally be represented by a Taylor series. We are thus approximating functions that are approximately polynomials of order 7 and the discontinuities cancel to a large degree.

Instead of calculating the exact matrix elements we therefore use matrix elements with respect to a smoothed version $\tilde{\phi}$ of the Daubechies scaling functions.

$$V_{i,j,k;i',j',k'} \approx \sum_{l,m,n} \tilde{\phi}_{i',j',k'}(\mathbf{r}_{l,m,n}) V(\mathbf{r}_{l,m,n}) \tilde{\phi}_{i,j,k}(\mathbf{r}_{l,m,n}) = \sum_{l,m,n} \tilde{\phi}_{0,0,0}(\mathbf{r}_{i'+l,j'+m,k'+n}) V(\mathbf{r}_{l,m,n}) \tilde{\phi}_{0,0,0}(\mathbf{r}_{i+l,j+m,k+n}) \; . \tag{78}$$

The magic filter ω is defined by

$$\omega_{l,m,n} = \tilde{\phi}_{0,0,0}(\mathbf{r}_{l,m,n})$$

The relation between the true functional values, i.e. the scaling function, and ω is shown in figure 14. Even though Eq. 78 is not a particularly good approximation for a single matrix

element it gives an excellent approximation for the expectation values of the local potential energy

$$\int dx \int dy \int dz\ \Psi(x,y,z)V(x,y,z)\Psi(x,y,z)$$

In practice we do not explicitly calculate any matrix elements but we apply only filters to the wave function expansion coefficients as will be shown in the following. This is mathematically equivalent but numerically much more efficient.

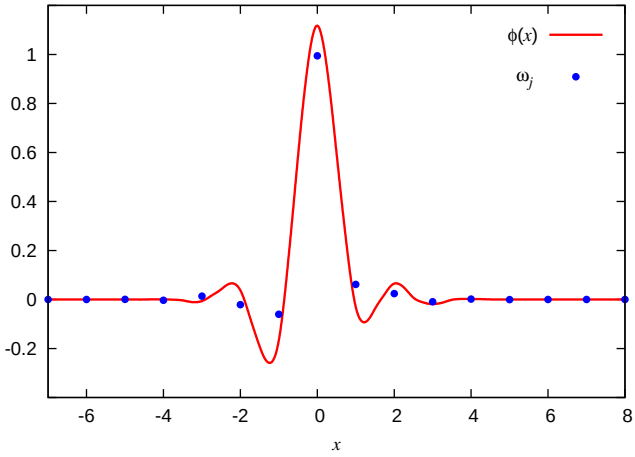


Figure 14. The magic filter ω_i for the least asymmetric Daubechies-16 basis. The values of the magic filter do not coincide with the functional values of the scaling function but represent the behavior of this function in some neighborhood.

Once we have calculated $\tilde{\Psi}_{i,j,k}$ the approximate expectation value ϵ_V of the local potential V for a wave function Ψ is obtained by simple summation on the real space grid:

$$\epsilon_V = \sum_{j_1,j_2,j_3} \tilde{\Psi}_{j_1,j_2,j_3} V_{j_1,j_2,j_3} \tilde{\Psi}_{j_1,j_2,j_3}$$

15.2 Treatment of the non-local pseudopotential

The energy contributions from the non-local pseudopotential have for each angular moment l the form

$$\sum_{i,j} \langle \Psi | p_i \rangle h_{ij} \langle p_j | \Psi \rangle$$

where $|p_i\rangle$ is a pseudopotential projector. When applying the hamiltonian operator on a wave function, such a separable term requires the calculation of

$$|\Psi\rangle \rightarrow |\Psi\rangle + \sum_{i,j} |p_i\rangle h_{ij} \langle p_j | \Psi \rangle \ .$$

It follows from Eq. 51 that the scaling function expansion coefficients for the projectors are given by

$$\int p(\mathbf{r})\phi_{i_1,i_2,i_3}(\mathbf{r})d\mathbf{r} \ . \tag{79}$$

The GTH-HGH pseudopotentials[14,15] have projectors which are written in terms of Gaussian times polynomials. This form of projectors is particularly convenient to be expanded in the Daubechies basis. Since a 3-dimensional Gaussian $\langle \mathbf{r}|\mathbf{p}\rangle = \mathbf{e}^{-\mathbf{cr^2}}\mathbf{x}^{\ell_x}\mathbf{y}^{\ell_y}\mathbf{z}^{\ell_z}$ is a product of three 1-dimensional Gaussians, the 3-dimensional integral 79 can be factorized into a product of three 1-dimensional integrals.

$$\begin{aligned}\int \langle \mathbf{r}|\mathbf{p}\rangle \phi_{\mathbf{i_1},\mathbf{i_2},\mathbf{i_3}}(\mathbf{r})\mathbf{dr} &= W_{i_1}(c,\ell_x)W_{i_2}(c,\ell_y)W_{i_3}(c,\ell_x) \ , \\ W_j(c,\ell) \quad &= \textstyle\int_{-\infty}^{+\infty} e^{-ct^2}t^\ell \phi(t/h-j)dt\end{aligned}$$

The 1-dimensional integrals $W_j(c,\ell)$ are calculated in the following way. We first calculate the scaling function expansion coefficients for scaling functions on a 1-dimensional grid that is 16 times denser. The integration on this dense grid is done by summing the product of the Gaussian and the smoothed scaling function that is obtained by filtering the original scaling function with the magic filter[12]. This integrations scheme based on the magic filter has a convergence rate of h^{14} and we gain therefore a factor of 16^{14} in accuracy by going to a denser grid. This means that the expansion coefficients are for reasonable grid spacings h accurate to machine precision. After having obtained the expansion coefficients with respect to the fine scaling functions we obtain the expansion coefficients with respect to the scaling functions and wavelets on the required resolution level by one-dimensional fast wavelet transformations (Eq. 26). No accuracy with respect to the scaling function coefficients on the lower resolution levels is lost in the wavelet transforms and our representation of the coarse scaling function coefficients of the projectors is therefore typically accurate to nearly machine precision.

16 Final Remarks

Even though wavelet basis sets allow for a very high degree of adaptivity, i.e. many levels of wavelets in Eq. 52, such a high degree of adaptivity causes some numerical overhead that slows down a program. For this reason we have adopted in the BigDFT electronic structure program (http://www-drfmc.cea.fr/sp2m/L_Sim/BigDFT/) only a low degree of adaptivity, namely two resolution levels which are obtained by a set of scaling function augmented by a set of 7 wavelets in the high resolution regions. In most cases, the more rapid variation of the wavefunction around in the chemical bonding region is described by scaling functions plus wavelets whereas the slower variation in the tail regions of the wavefunction is described by scaling functions only. This is typically sufficient since pseudopotentials are used to eliminate the strongly varying core electrons and to account for relativistic effects. All electron wavelet based electronic structure programs do however exist as well[16,17].

References

1. F. Gygi, *Europhys. Lett.* **19**, 617 (1992); Phys. Rev. B **48** 11692 (1993); *Phys. Rev.* B **51** 11190 (1995).
2. Y. Meyer, *"Ondelettes et opérateurs"* Hermann, Paris, 1990.
3. I. Daubechies, *"Ten Lectures on Wavelets"*, SIAM, Philadelphia (1992).
4. S. Goedecker: "Wavelets and their application for the solution of partial differential equations", Presses Polytechniques Universitaires et Romandes, Lausanne, Switzerland 1998, (ISBN 2-88074-3 98-2)
5. W. Sweldens and R. Piessens, *"Wavelets sampling techniques"*, Proceedings of the Joint Statistical Meetings, San Fransisco, August 1993.
6. W. Sweldens, *Appl. Comput. Harmon. Anal.* **3**, 186 (1996).
7. G. Deslauriers and S. Dubuc, *Constr. Approx.* **5**, 49 (1989).
8. C. Tymczak and X. Wang, *Phys. Rev. Lett.* **78**, 3654 (1997).
9. G. Beylkin, *SIAM J. on Numerical Analysis* **6**, 1716 (1992).
10. S. Goedecker, O. Ivanov, *Sol. State Comm.* **105** 665 (1998).
11. Luigi Genovese, Thierry Deutsch, Alexey Neelov, Stefan Goedecker and Gregory Beylkin, J. Chem. Phys. **125**, 074105 (2006)
12. A. I. Neelov and S. Goedecker, J. of. Comp. Phys. **217**, 312-339 (2006)
13. Luigi Genovese, Alexey Neelov, Stefan Goedecker, Thierry Deutsch, Alireza Ghasemi, Oded Zilberberg, Anders Bergman, Mark Rayson and Reinhold Schneider, J. Chem. Phys. **129**, 014109 (2008)
14. S. Goedecker, M. Teter, and J. Hutter, *Phys. Rev. B* **54**, 1703, (1996).
15. C. Hartwigsen, S. Goedecker and J. Hutter, Phys. Rev. B **58**, 3641 (1998)
16. T.D. Engeness and T.A. Arias, Phys. Rev. B65, 165106-1 (2002)
17. Robert J. Harrison, George I. Fann, Takeshi Yanai, Zhengting Gan, and Gregory Beylkin J. Chem. Phys. **121** 11587 (2004)